IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alejandro Cuartas Villada
October 20, 2021

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

To compete successfully with SpaceX in space travel, SpaceY's executive leadership team must understand the factors behind SpaceX's successes.

A large part of SpaceX's success can be attributed to its ability to land and recover the "first stage" component of its launch vehicles. By successfully recovering the first stage component, SpaceX is able to launch additional missions at significantly lower costs relative to competitors.

Based on the importance first stage recovery (or successful landing) as a cost driver in space launch, we conducted an in-depth analysis of SpaceX's historical launch campaigns based on available data to propose a machine learning model capable of predicting whether a launch will be successful with 84% accuracy.

Introduction

Our task was to develop a model capable of predicting whether future SpaceX launches will be successful, defined as successful recovery of the first stage in a launch.

To gather data, we gathered SpaceX launch data from the SpaceX REST API and by deploying Python-based web scrape methods to obtain Falcon 9 launch records.

We conducted exploratory analysis using SQL and also by using Pandas and Matplotlib. To further enhance the visualization analytics, we deployed a web-based dashboard using Plotly Dash to analyze launch records and built an interactive map to derive insights about launch site and proximities to geographical areas.

The last step of our exercise was to deploy a machine learning module capable of predicting whether a launch will be successful based on factors such as orbit type, payload mass, launch site, and other relevant factors.

Section 1

Methodology

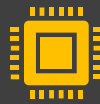
Methodology: Pt 1

Data collection methodology:

Data was collected by using the **SpaceX REST API** and **Web scraping** HTML tables containing Falcon 9 launch records

Data wrangling

We calculated the number of launches on each site, number/occurrences of each orbit, mission outcomes per orbit type and normalizing outcome labels as “successful” and “not successful”



Perform exploratory data analysis (EDA) using visualization and SQL

DB2 instance was created on IBM Watson Studio.

SpaceX dataset loaded into DB2 was analyzed through SQL magic commands in Python

Methodology: Pt 2

Folium and Plotly Dash

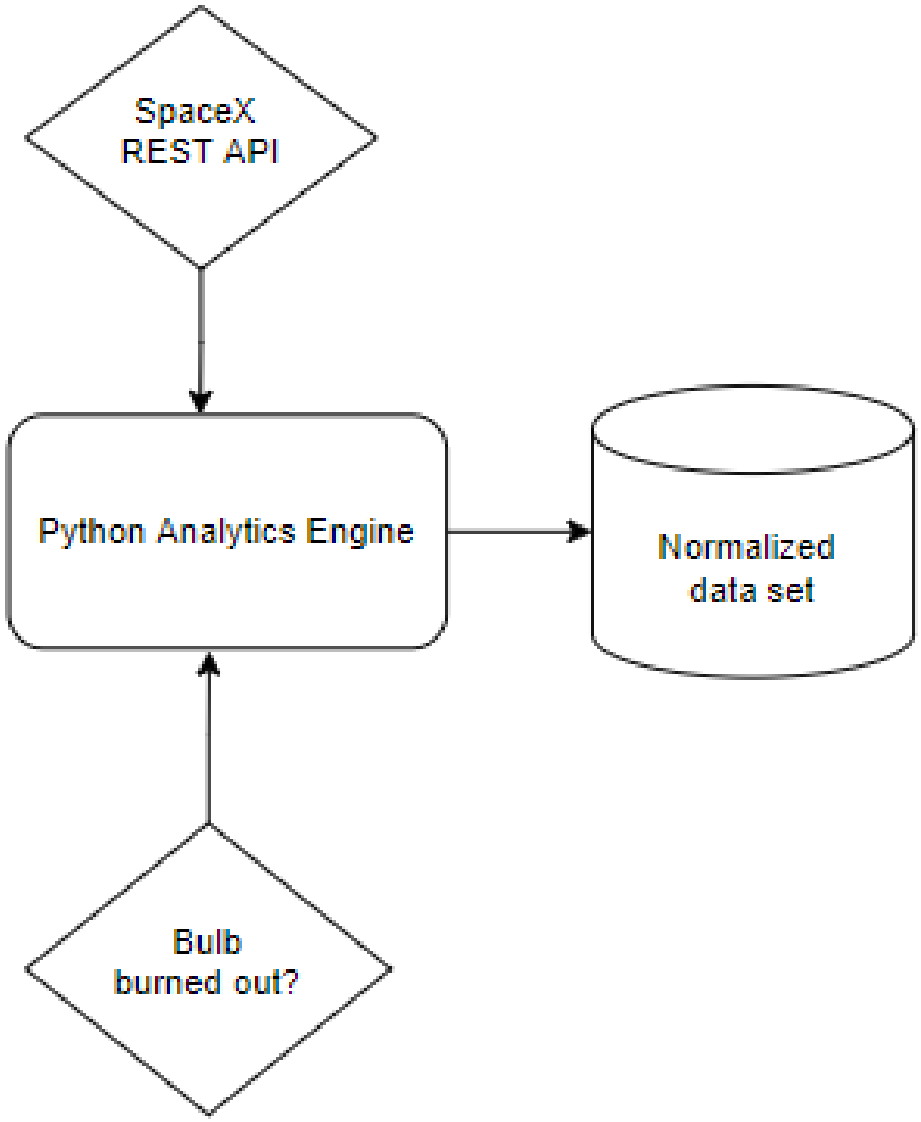
- We marked all launch sites using Folium map and also marked the success/failed launches for each site.
- We calculated the distances between a launch site to its proximities and set up the interactive dashboard to graph various relationships between success outcomes, payload, launch sites, and launch site proximities.

Predictive Analysis

- We standardized data and split data into a train set and test set.
- We tested different machine learning models using Logistic Regression, SVM and Classification Trees to find the most accurate method of doing predictive analysis.

Data Collection & Data Wrangling

- We collected data into our Python Jupyter notebook through two primary methods:
  - **SpaceX REST API;** and,
  - **Web scraping** HTML tables containing Falcon 9 launch records
- Once we gathered the data, we normalized the data on Python into an acceptable DB format to conduct further visual and SQL analysis.
- **GitHub Links:**
  - **SpaceX REST API**
  - **Web scraping exercise**
  - **Data Wrangling**



Data Collection SpaceX API

API Response

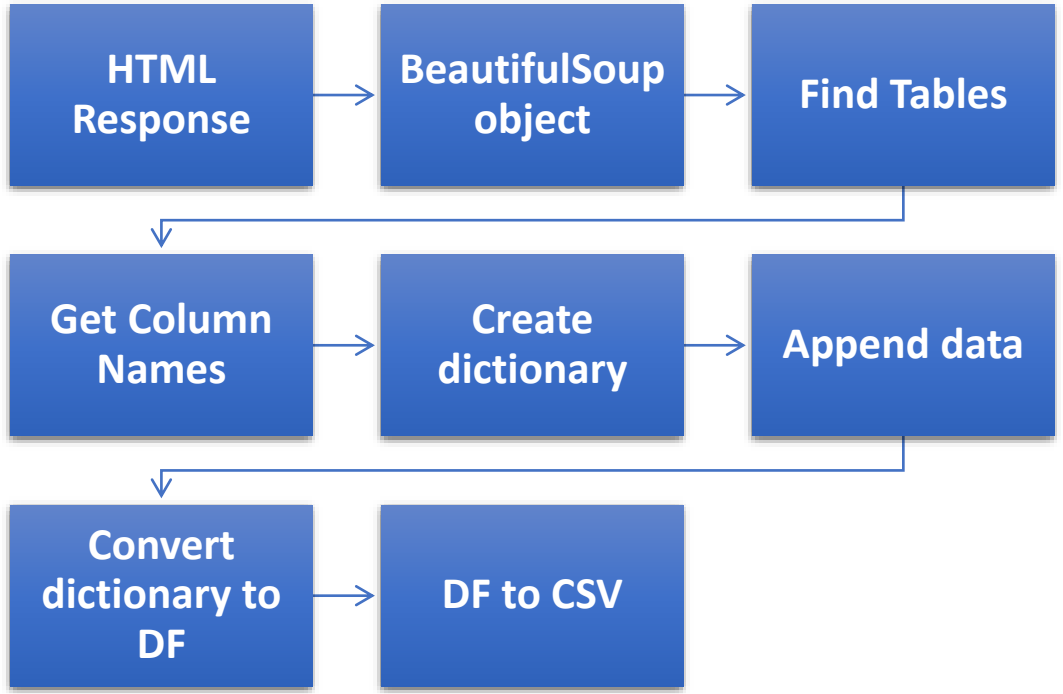
JSON conversion

List to dictionary & DF

Filter DF

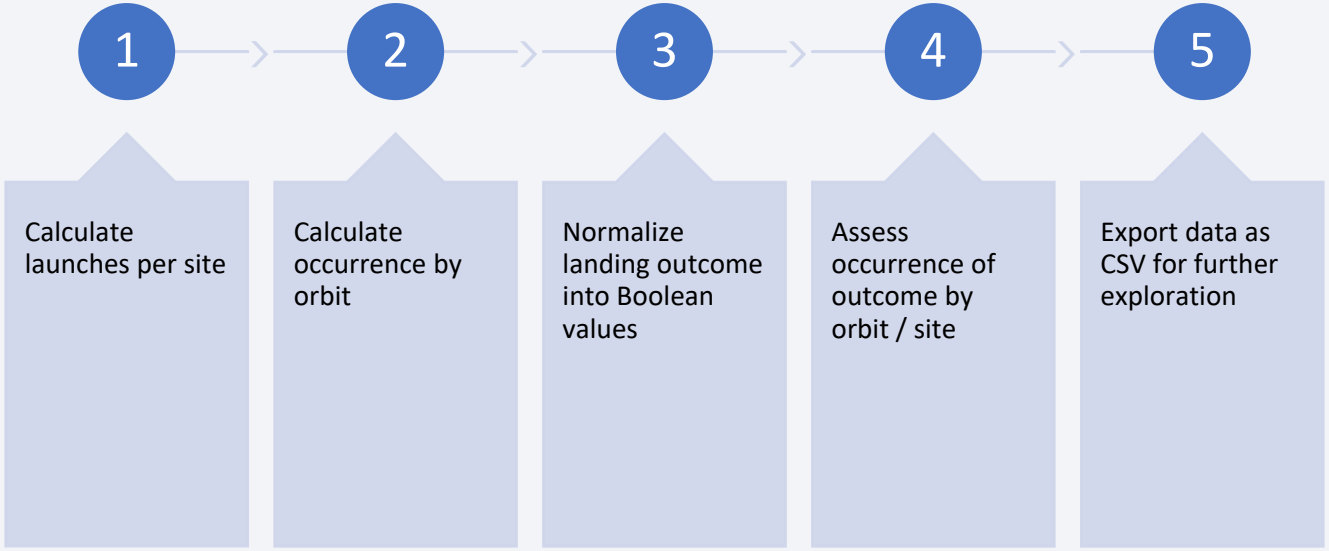
Export CSV

Data Collection Web Scrapping



Data Wrangling

- Our dataset analysis found several cases where a booster did not land successfully.
- The reasons for failure differed and, in some cases, “success” or “failure” was measured in different contexts.
- The data wrangling portion of our analysis focused on converting the variety of outcomes into a Boolean 1/0 format to determine successful vs. unsuccessful.
- GitHub URL: [Data Wrangling Lab]



EDA with Data Visualization

- We used scatter plots to understand trends such as:
  - Flight # vs Pay load Mass
  - Relationship between Flight # and Launch Site
  - Relationship between Launch Site and Pay Load
  - Successful launches by orbit type
- We used the bar chart to visualize the success of each orbit type and spot ones with high success rates.
- The line chart allowed us to visualize average success rate by year, indicating success rate increased to over 60% by 2020.

GitHub URL

EDA with SQL

GitHub URL

We executed 10 different SQL queries

- Names of each unique launch site
- 5 records where launch sites started with “CCA”
- Total payload mass carried by boosters launched by NASA(CRS)
- Avg. payload mass carried by booster F9 v1.1
- Date when the first successful landing outcome in ground pad was achieved
- List the names of boosters which have success in drone ship and have payload masses between 4000 & 6000kg
- List the total number of successful and failure mission outcomes
- List the name of the booster versions which carried the maximum payload mass.
- List failed landing outcomes in drone ship by booster and site
- Rank the count of landing outcomes between June 4, 2010 and March 20, 2017 in descending order

Build an Interactive Map with Folium

- To visualize the Launch Data into an interactive map, we used latitude and longitude coordinates at each launch site and added circle markers and labels to distinguish them.
- We assigned the dataframe launch outcomes to classes with Green and Red markers to visualize success rate by location.
- Haversine's formula used to calculate distance from the Launch Site to various landmarks to find possible correlation between location and success rates. **Lines** were drawn to measure distance to landmarks.
- GitHub URL

Predictive Analysis (Classification)

We built our models in NumPy and Pandas by splitting data into training and test sets. We decided

We evaluated each model by running accuracy tests and plot confusion matrix.

To improve the model, we used algorithm tuning and feature engineering codes

The model with the best accuracy was ultimately selected as the best performing classification model.

GitHub URL

Results

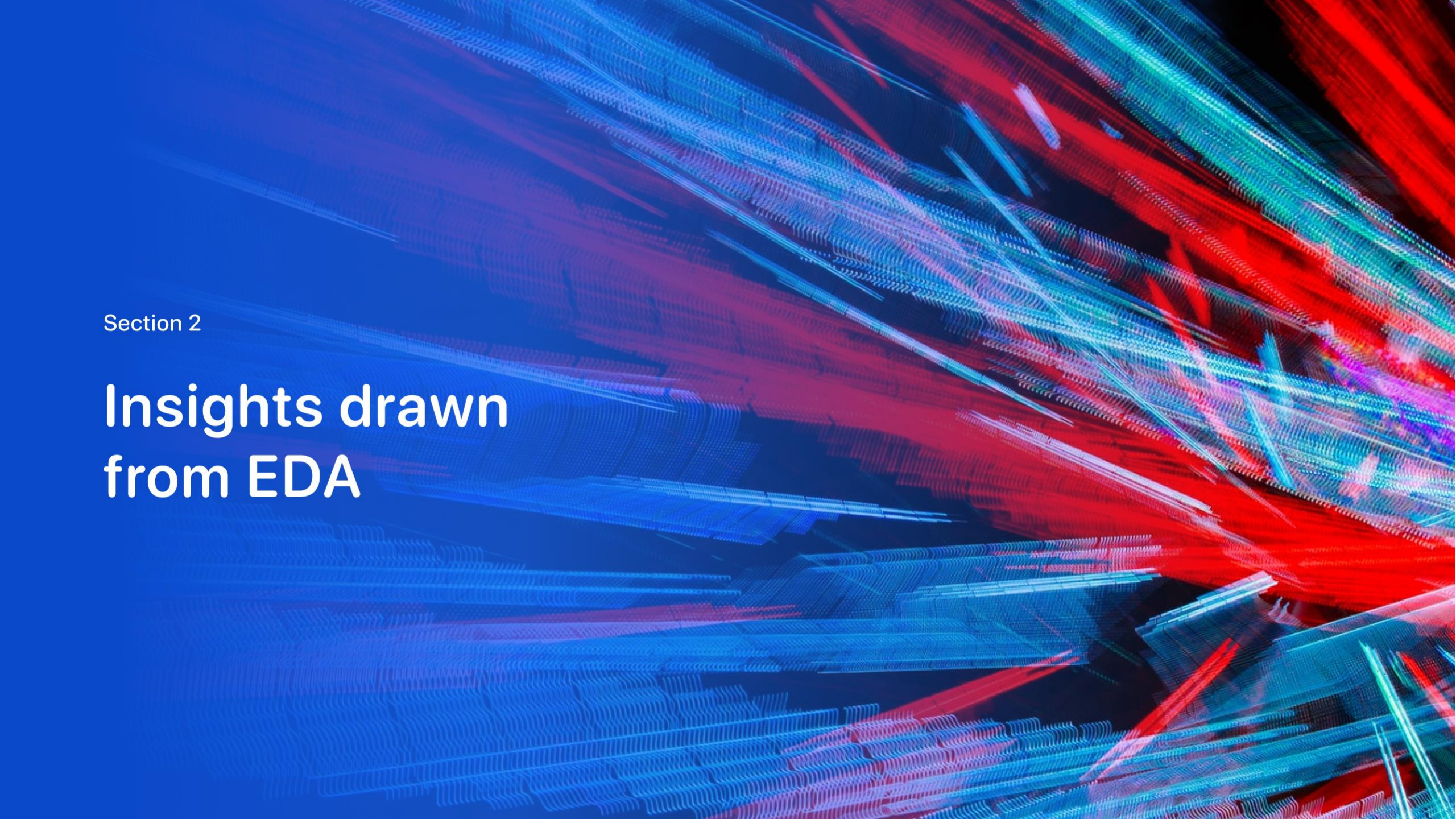
EXPLORATORY DATA ANALYSIS RESULTS

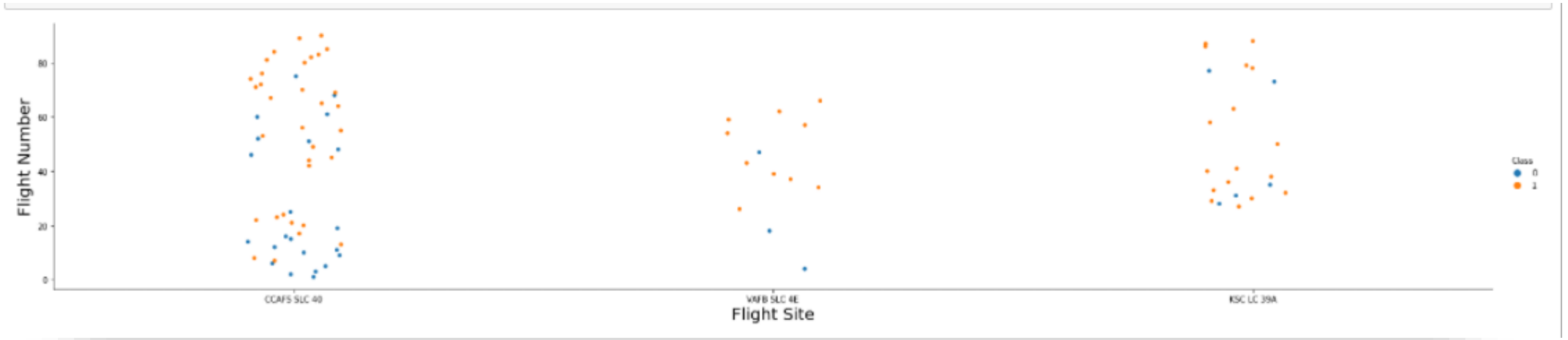
INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

PREDICTIVE ANALYSIS RESULTS

Section 2

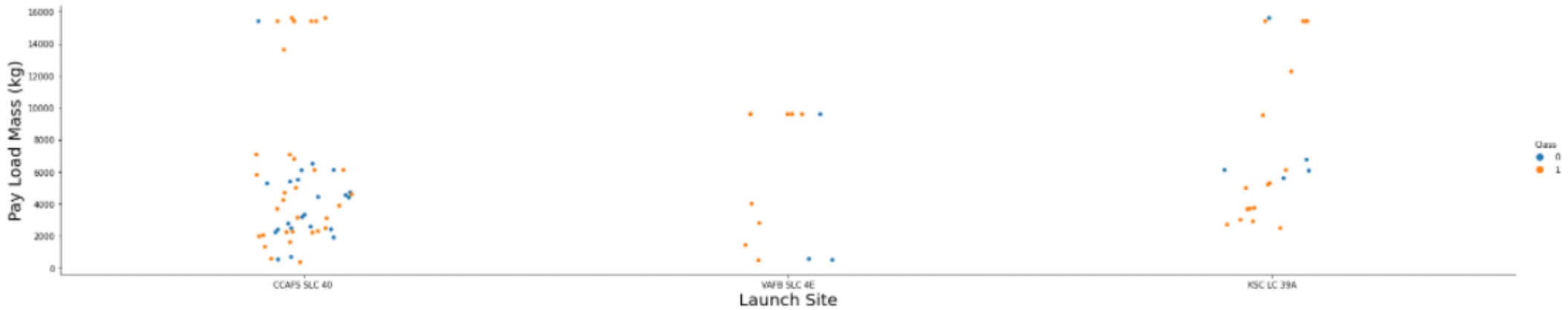
Insights drawn from EDA





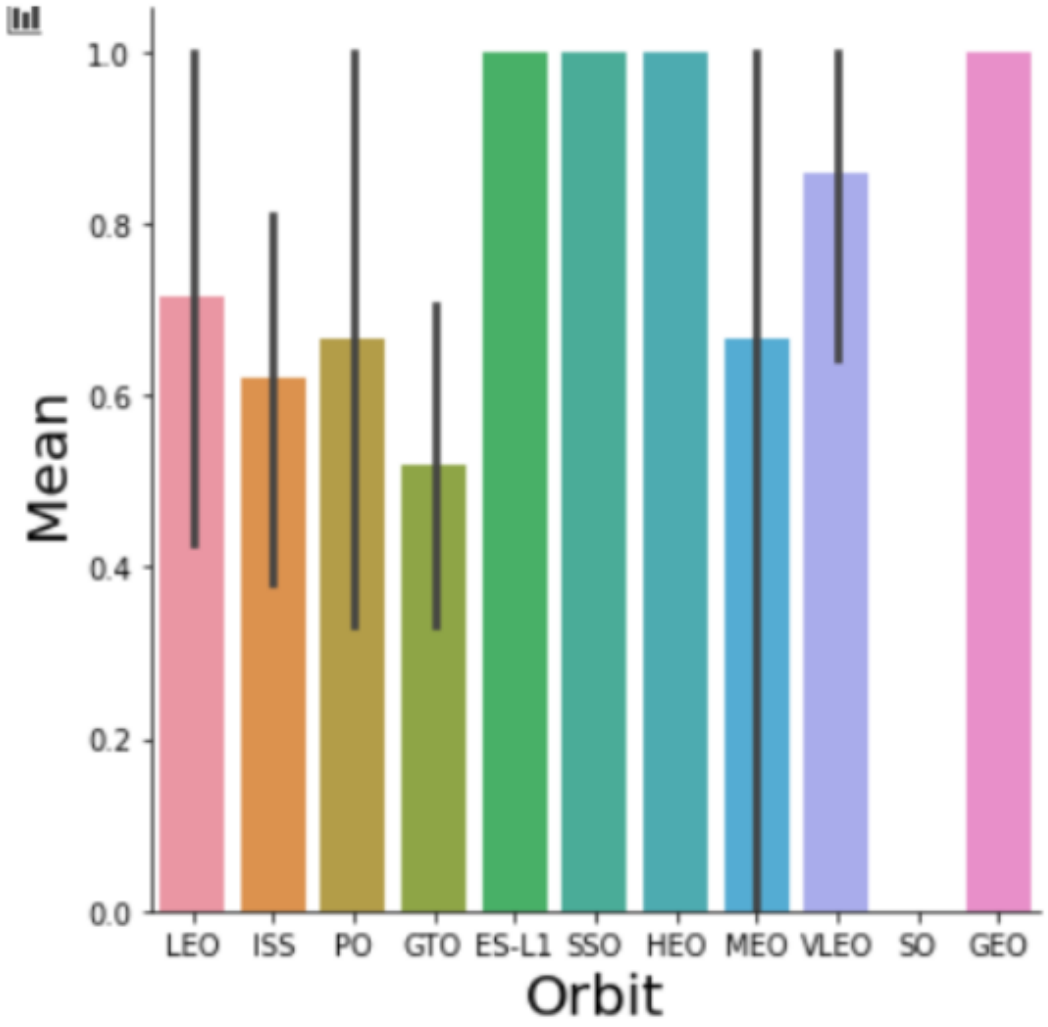
Flight Number vs. Launch Site

- Different launch sites have different success rates and VAFB has the highest (77%) success rate. We can also observe earlier flights as characterized by earlier flight numbers had more fail rates than more recent (higher count) flight numbers.



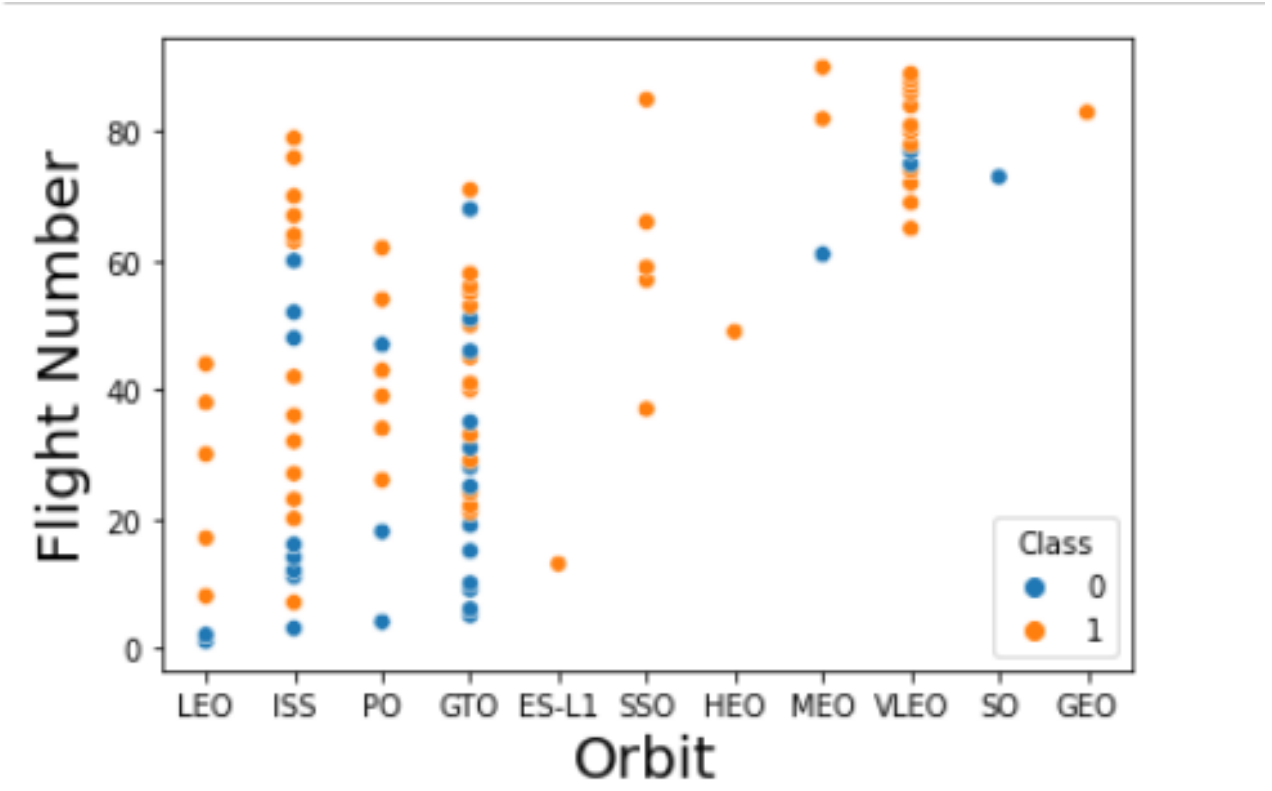
Payload vs. Launch Site

- VAFB site has no logged heavypayload launches (above 10,000kg) which may partly explain this site's higher success rate. However, we can also see CCAFS had many failures in the sub-10,000KG payload class which we may attribute to launch attempts in earlier years when the technology was less advanced.



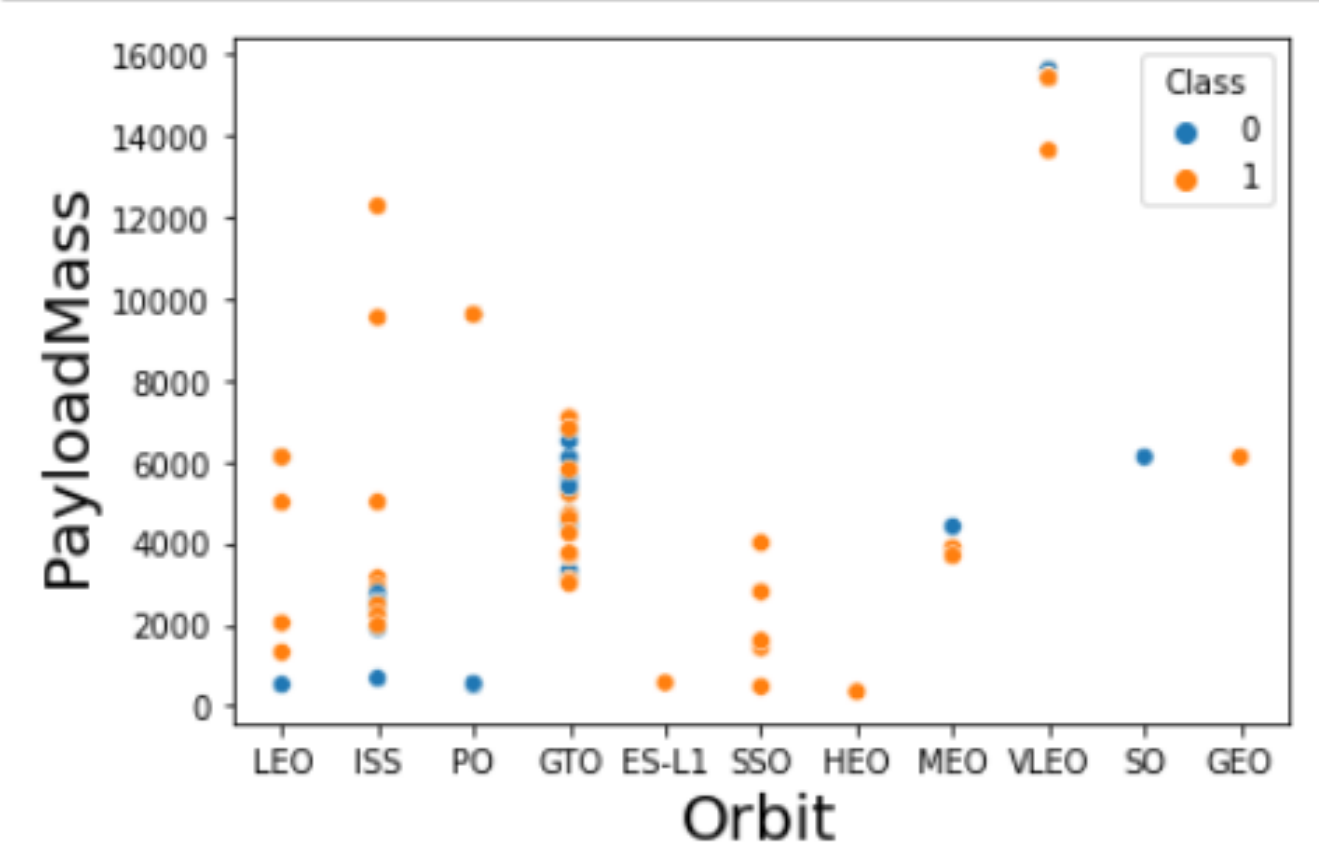
Success Rate vs. Orbit Type

- Some orbits have higher success rate. GEO, HEO, SSO, and ES-L1 have the best success rates at 100%. Conversely, GTO, ISS, MEO, and PO have lower success rates ranging from ~45% to 60%.



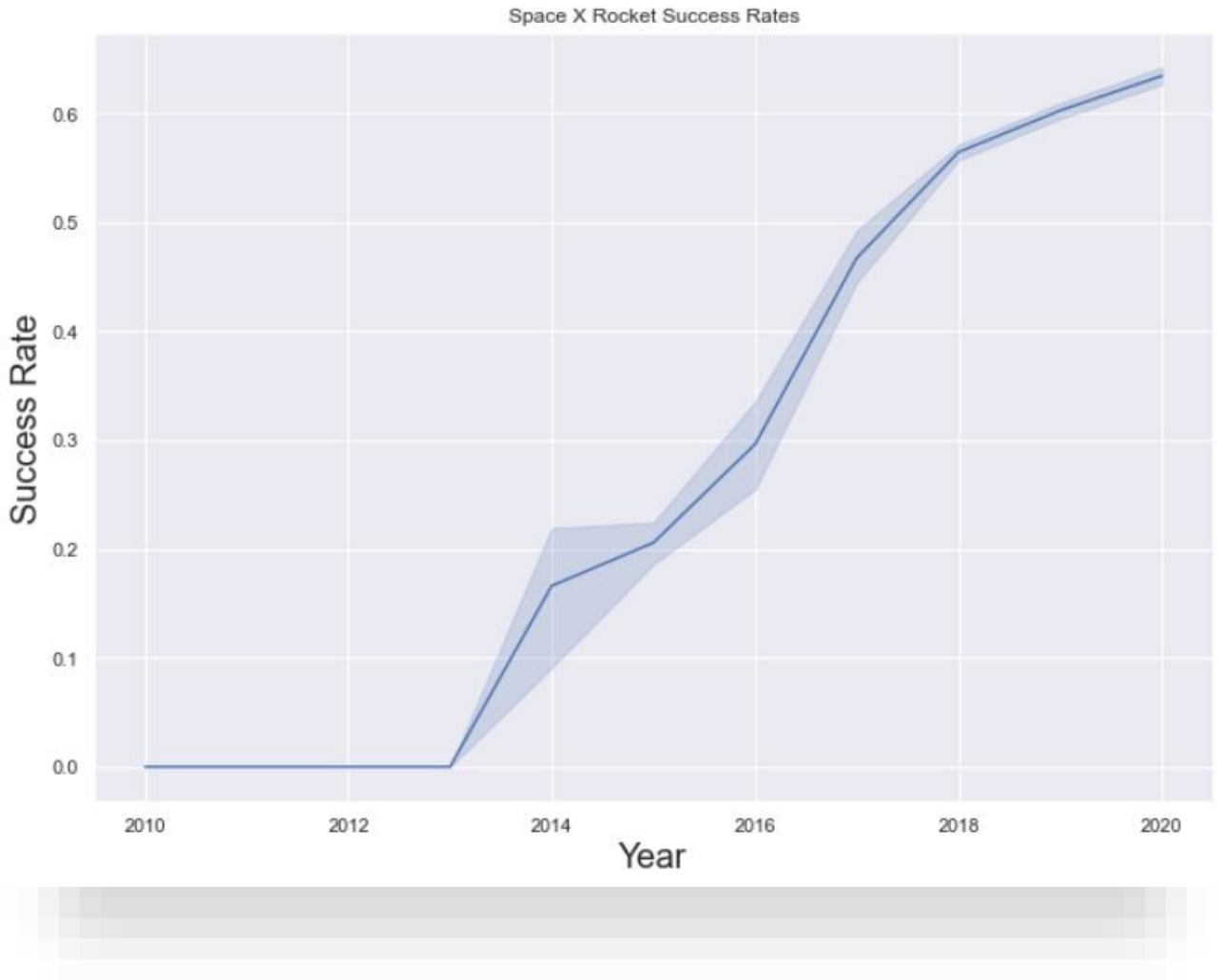
Flight Number vs. Orbit Type

- In LEO orbit, success appears related to the number of flights. However, this does not seem to be the case with other orbit types such as GTO and ISS. We can also see many recent successful launches have been in VLEO orbit.



Payload vs. Orbit Type

- Heavy payloads (greater than 10,000kg) are limited to ISS, PO and VLEO with relatively high success rate. We also see GTO range limits from 4,000kg – 8,000kg with varying success rates.



Launch Success Yearly Trend

- Success rate from 2013 to 2020 increased significantly to above 60% success rate. This indicates that, on average, successful launches have become the norm.

EDA with SQL

SQL Queries Explained

```
%sql select distinct(LAUNCH_SITE) from SPACEXDATASET
```

```
 * ibm_db_sa://nnn69380:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

All Launch Site Names

- We used **DISTINCT** to pull only unique values in the **Launch_Site** column from the table named **SpaceXDataSet**

```
%sql select * from SPACEXDATASET where LAUNCH_SITE like 'CCA%' limit 5
```

```
 * ibm_db_sa://nnn69380:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Launch Site Names Begin with 'CCA'

- We used a **WHERE** condition to define that we wanted to query for **Launch_Site** starting with “CCA%” by using the **LIKE** function
- Select * ensures we pulled all relevant columns for the records that satisfied our condition.

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXDATASET WHERE CUSTOMER = 'NASA (CRS)'
```

```
 * ibm_db_sa://nnn69380:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

1
45596

Total NASA(CRS)

- We used the **SUM aggregate** function and **WHERE** clause to calculate the total payload mass associated with NASA(CRS) boosters.

```
%sql SELECT avg(PAYLOAD_MASS__KG_) FROM SPACEXDATASET WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://nnn69380:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

1
2928

Average Payload Mass (F9 v1.1)

- We used the **AVG aggregate** function and **WHERE** clause to calculate the average payload mass associated with F9 v1.1 booster types.

```
%sql select min(date) from SPACEXDATASET where landing__outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://nnn69380:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

1
2015-12-22

First successful Landing Outcomes in ground pad

- We used the **MIN aggregate** function and **WHERE** clause to pull the first date on which there was a “Success (ground pad)” outcome.
- The **min** function, in this case, represents the earliest date on which our condition was satisfied.

```
%sql select booster_version\
from SPACEXDATASET where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ > 4000 and payload_mass__kg_ < 6000
```

```
 * ibm_db_sa://nnn69380:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Boosters with drone ship success and payload masses between 4000 and 6000kg.

- To pull a list of boosters with successful drone ship outcomes which were between 4,000 and 6,000kg, we used a **WHERE** clause with various conditions by using the **AND** statement in the query.

```
%%sql select MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXDATASET
GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://nnn69380:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

mission_outcome	total_number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Total number of successful and failure mission outcomes

- To pull the total number of successful and failure mission outcomes, our query selects two components (mission_outcome) and count(mission_outcome) with the **GROUP BY** mission_outcome. This provides a summary table of how many outcomes were associated with each mission_outcome category type.

```
%sql select BOOSTER_VERSION from SPACEXDATASET where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXDATASET)
```

```
 * ibm_db_sa://nnn69380:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

Booster versions which have carried the maximum payload pass

- To see which booster versions carried the maximum payload mass, it was necessary to use a **SUBQUERY.**
- In the **SUBQUERY,** we pull the maximum payload mass using the **MAX** function. Then, our query asks to pull all booster versions from the dataset where the payload mass KG is equal to the **MAX** calculated in our **SUBQUERY.**

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXDATASET
WHERE Landing__Outcome = 'Failure (drone ship)'
    AND YEAR(DATE) = 2015;
```

```
 * ibm_db_sa://nnn69380:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Failed landing outcomes in drone ship, booster version and launch site

- To see booster versions and launch sites associated with failed landings in drone ship, we used **SELECT** combined with a **WHERE** statement.
- In our **WHERE** statement, we stipulated **YEAR(DATE)** as 2015 to pull records specific to that year.

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXDATASET
WHERE Landing__Outcome = 'Failure (drone ship)'
    AND YEAR(DATE) = 2015;
```

```
 * ibm_db_sa://nnn69380:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

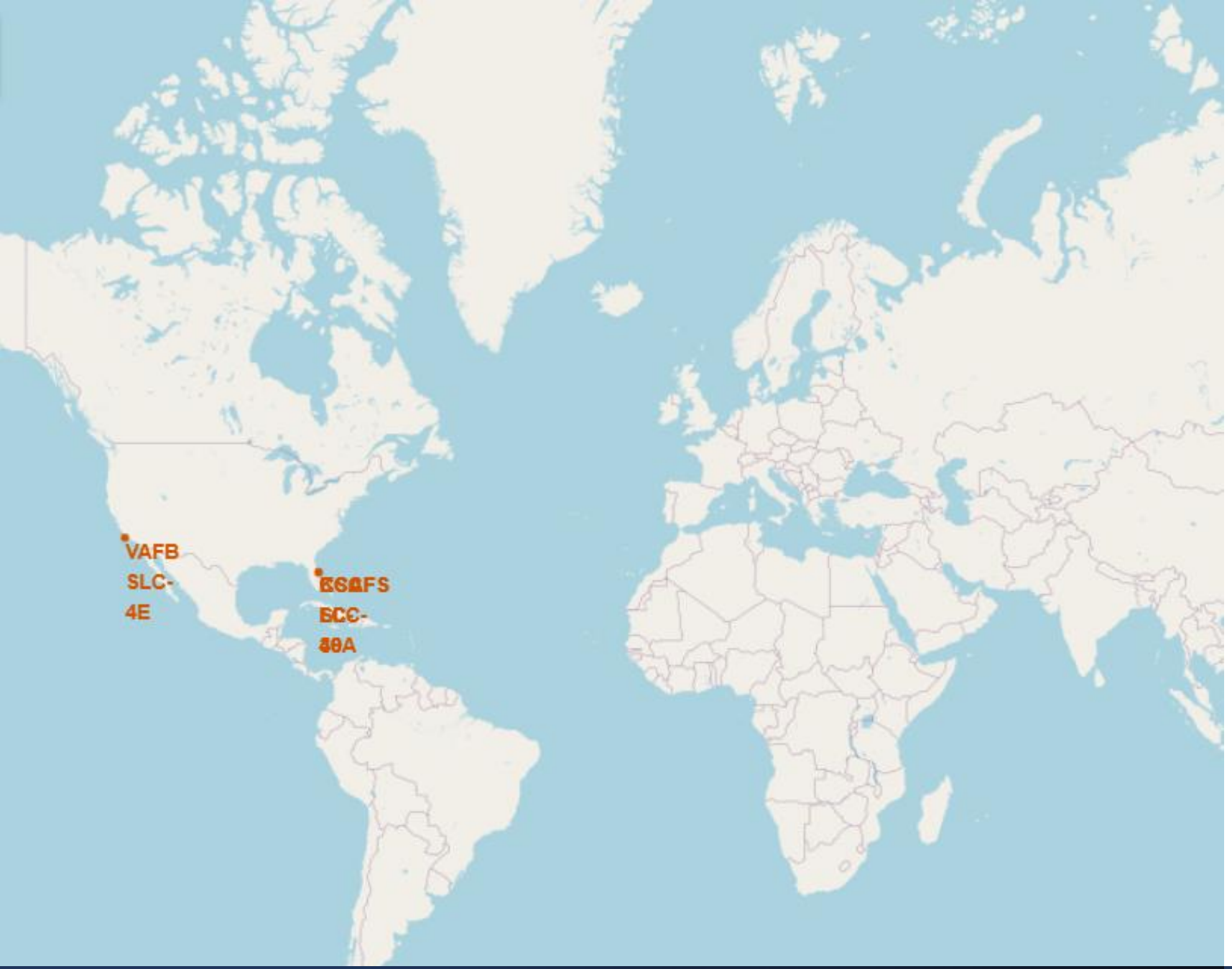
landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Rank count of landing outcomes between June 4, 2010 and March 20, 2017 in descending order

- We used a combination of **SELECT, WHERE, GROUP BY and ORDER BY** functions in our query to stipulate specific conditions, group the results by “landing_outcome”, and order by count in descending order.

Section 4

Launch Sites Proximities Analysis



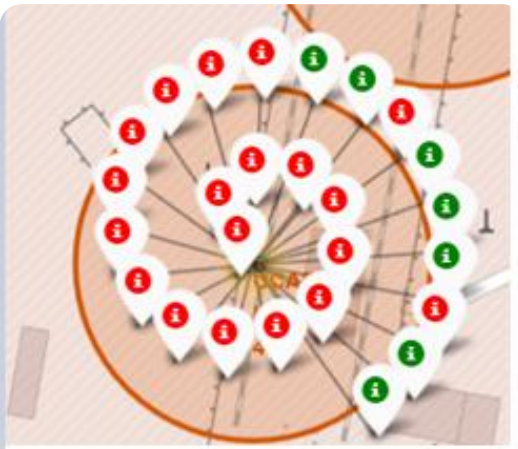
All launch sites

- We see that launch sites are located on a coast with close proximity to the water. This can manage risk associated with launches that fail for any reason.

Launch Outcomes by Sites

- Different launch sites have different success rates. KSC LC-39A is the most successful site as shown by the higher green-to-red ratio of pin points.

VAFB SLC-4E (California)



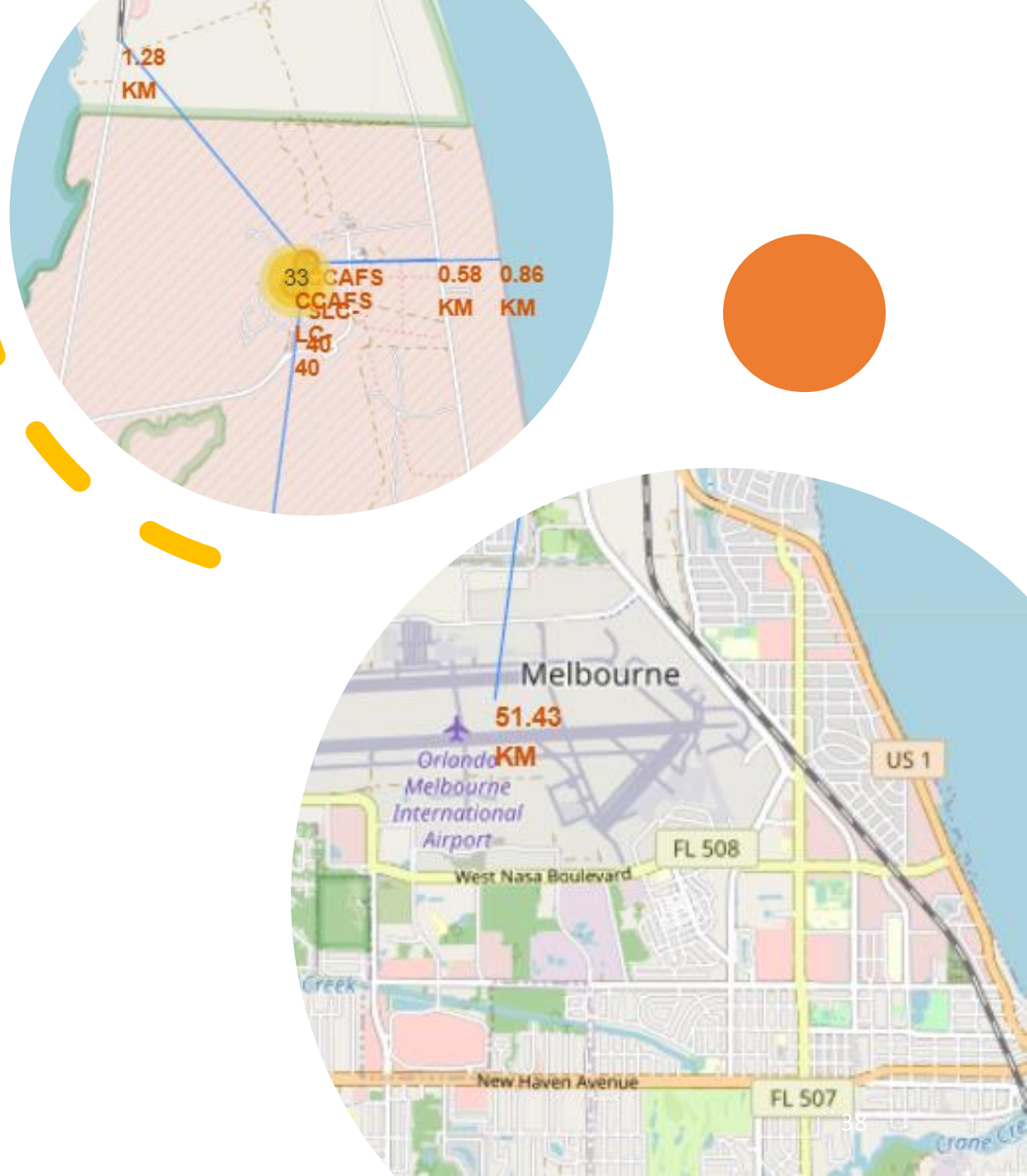
CCAFS LC-40 (Florida)

CCAFS SLC-40 (Florida)

KSC LC-39A (Florida)

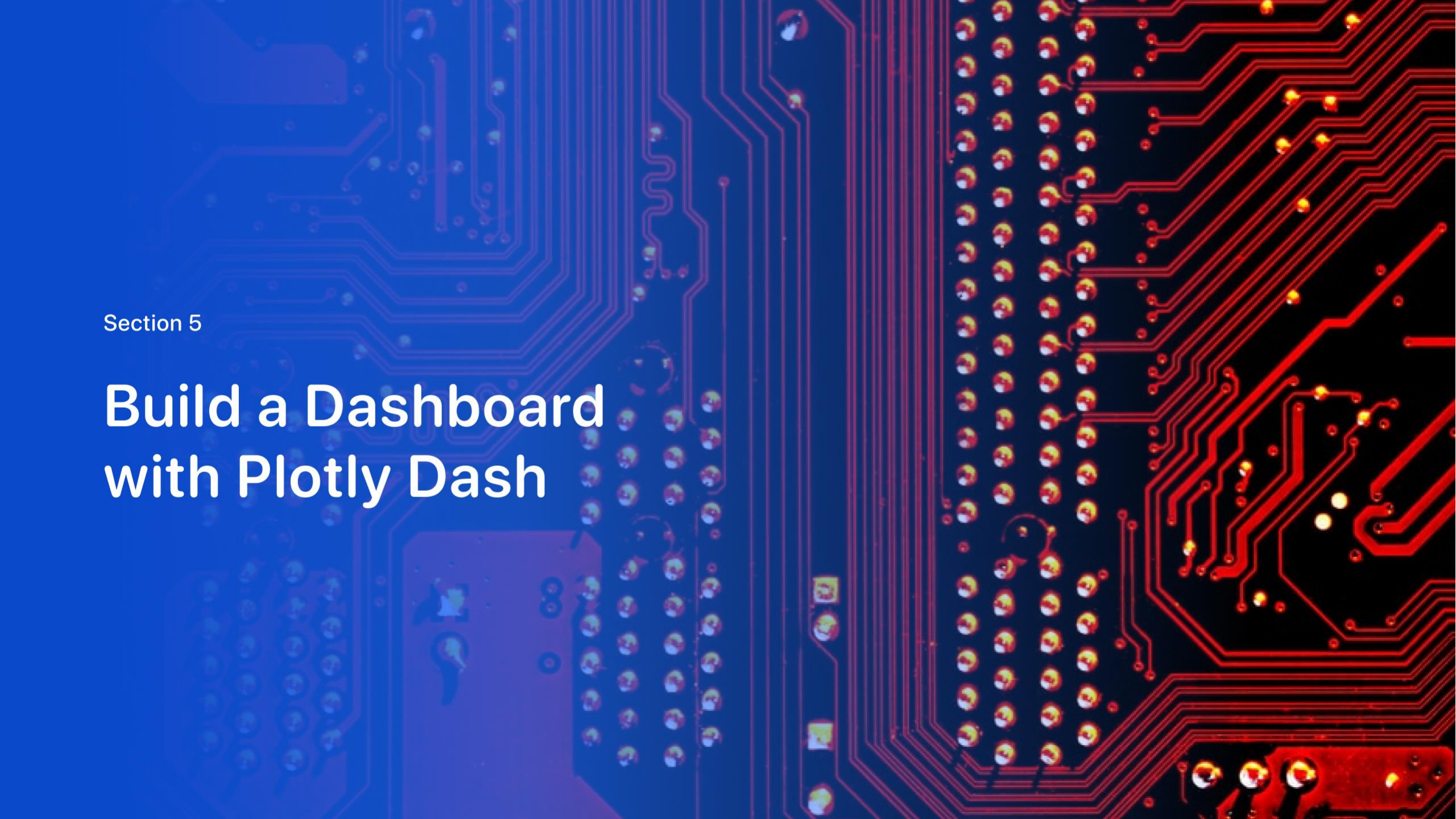
Landmark Proximities

- Sites have close proximity to the coast, highway and railroad and far proximity to major cities for safety. California site is also far away from a highway which is much safer in the event of failed launches.



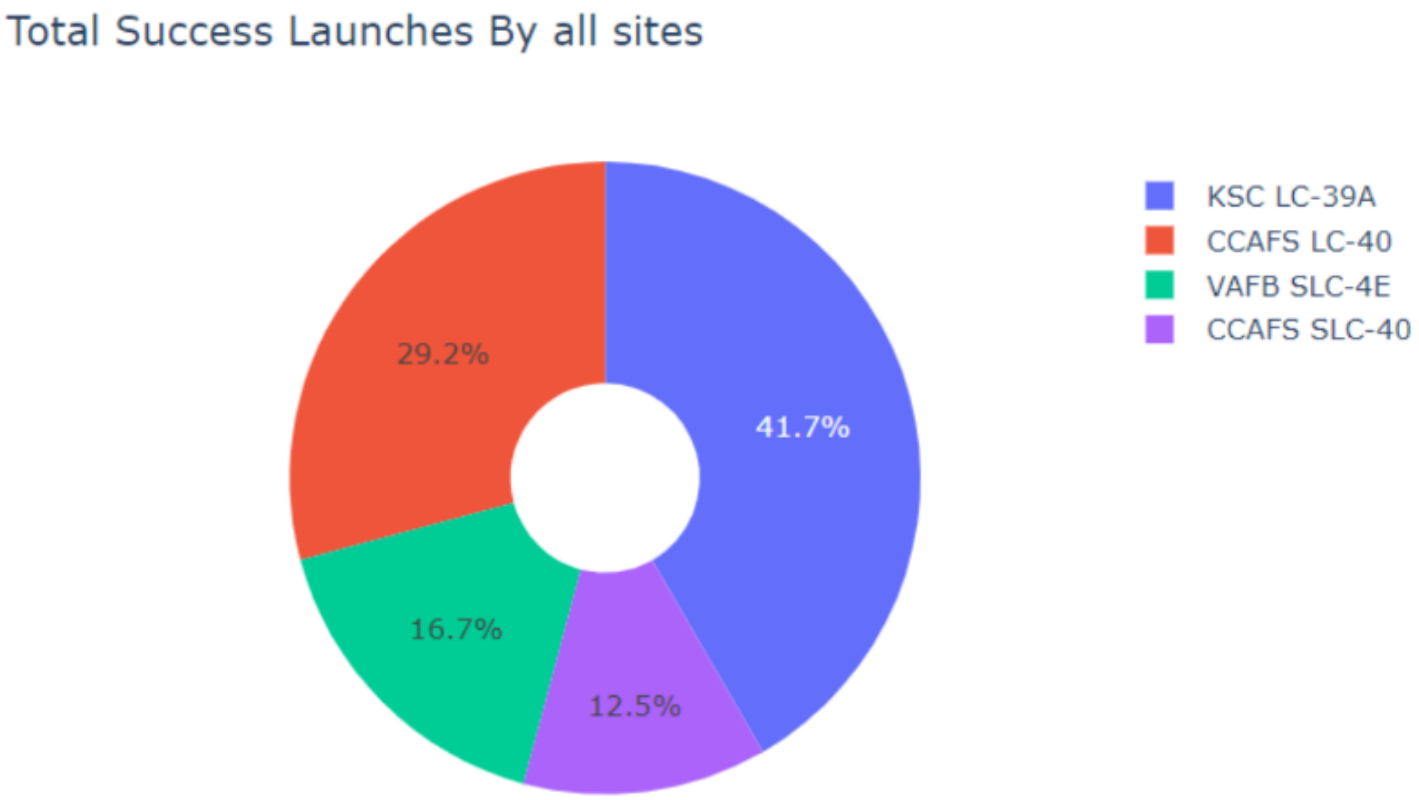
Section 5

Build a Dashboard with Plotly Dash



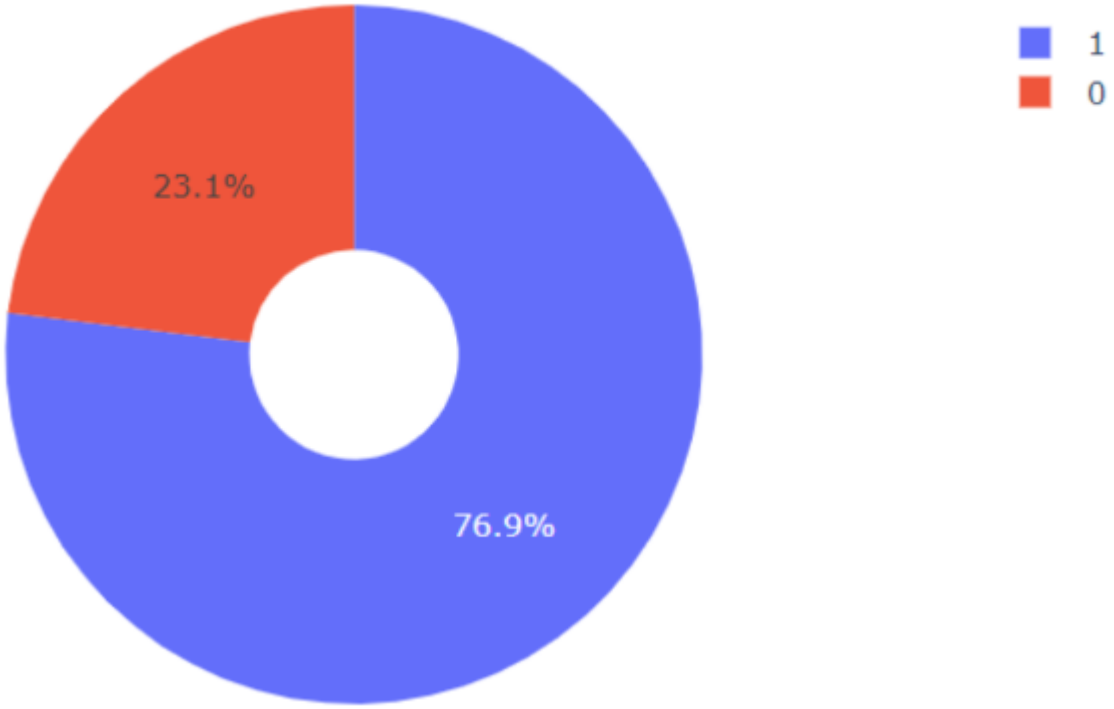
Success Launches by Sites

- Most successful launches originated from KSC LC-39A. Conversely, CCAFS SLC-40 had the fewest successful launches.



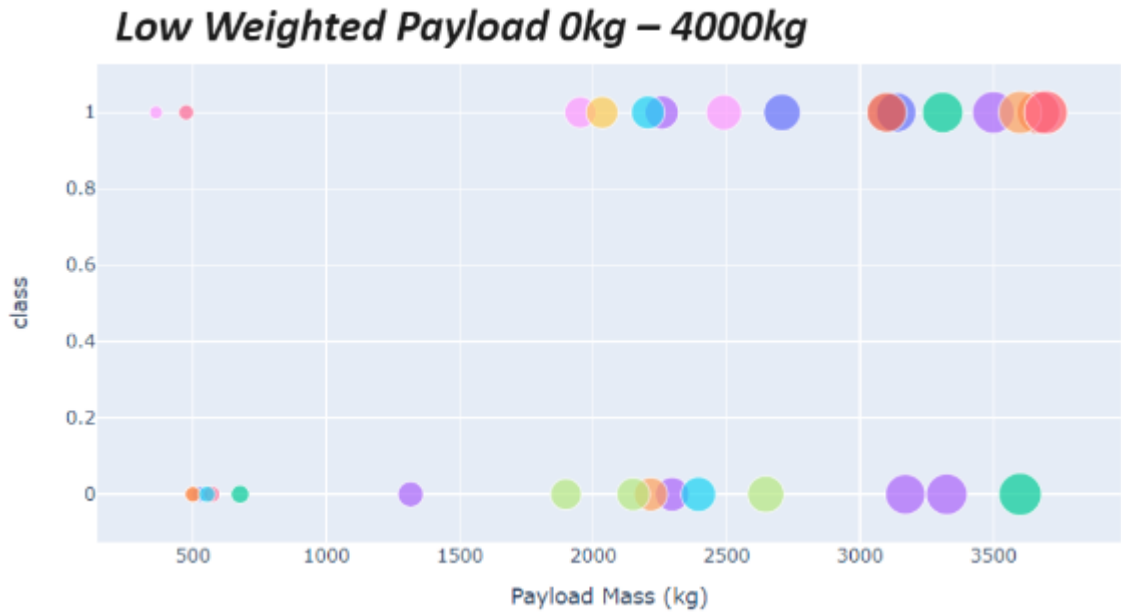
Launch Site with Highest Success Ratio

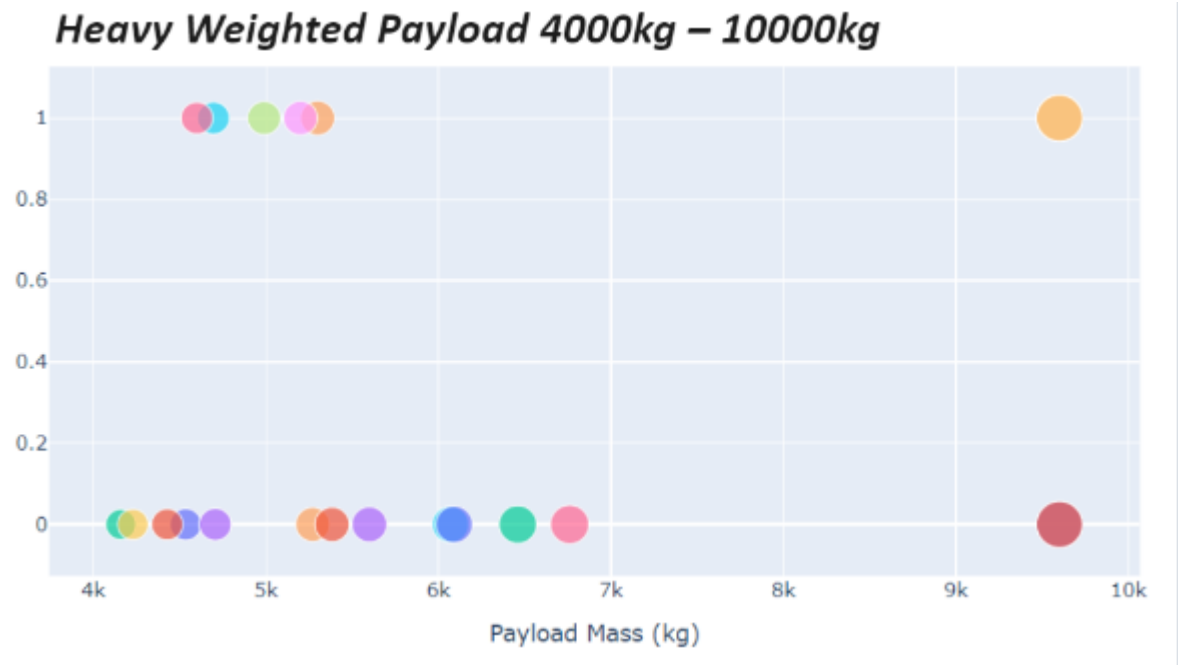
- KSC LC-39A had the highest success ratio, with a success rate of 77% and failure rate of 23%.



Payload vs Launch Outcomes

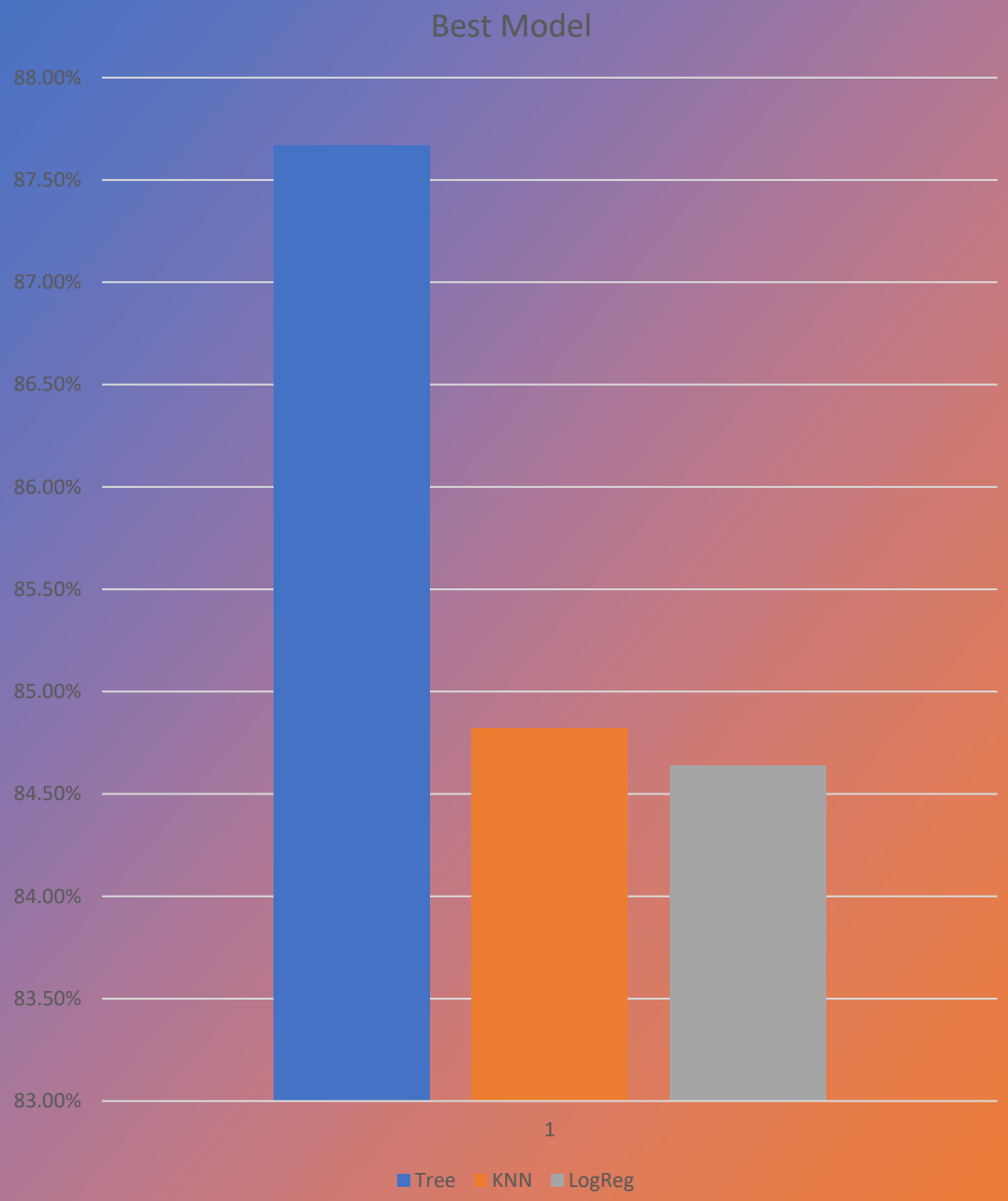
- We see much higher success rate with low-weighted payloads (Defined as 0-4000kg). Conversely, higher failure rate is associated with heavier payloads.





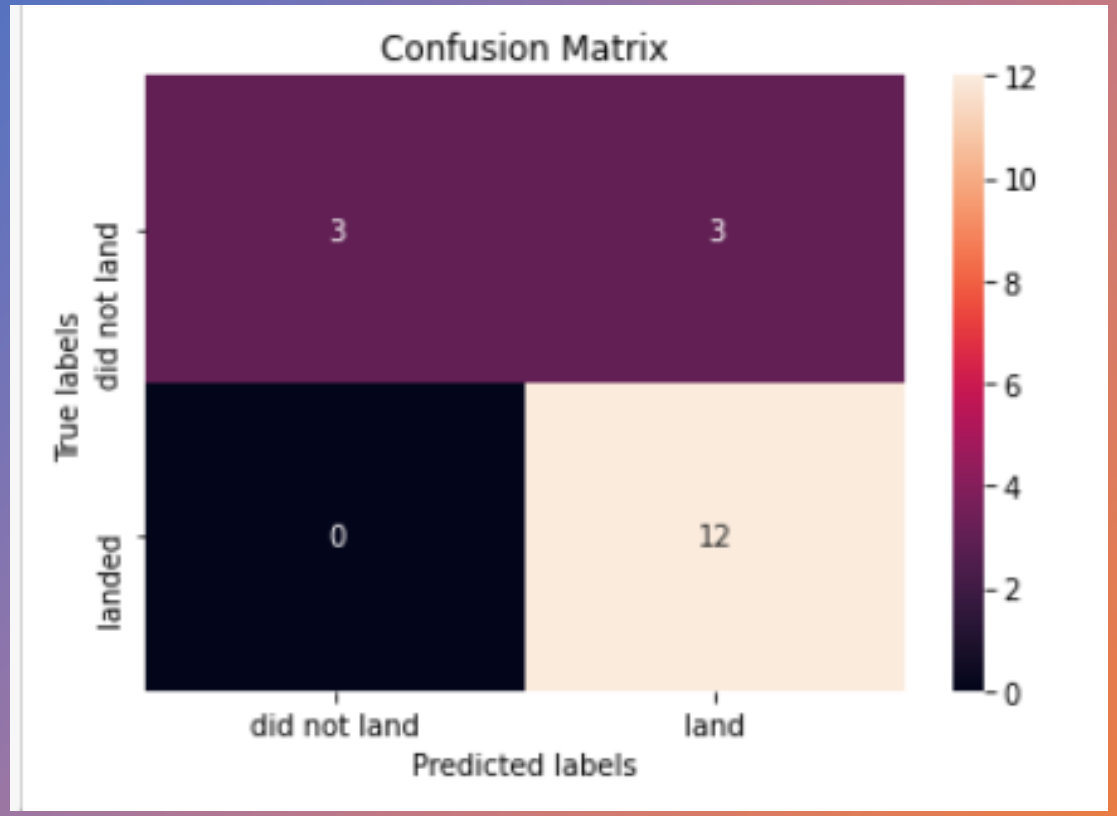
Section 6

Predictive Analysis (Classification)



Classification Accuracy

- While all models showed accuracy above 80%, the Tree Model stood out as the best model with accuracy above 87%.



Confusion Matrix

- All of our confusion matrices exhibited a similar pattern where the major issue was false positives.

Conclusions

- Since 2010, SpaceX has drastically improved its success rate to over 60%. As a result, it can be expected that SpaceX will be able to continue to benefit from low costs based on its proven ability to recover first stage during a launch.
- We can increase the odds of a successful launch by deploying sub-4000kg (low weight) payloads in orbit types GEO, HEO, SSO, and ES-L1. Conversely, we can minimize the risk of a failed launch by not using GTO, ISS, MEO, and PO orbit types.
- While our machine learning algorithm should continue to be developed to avoid false positives, we can say at this time that using the Tree Classifier Algorithm will produce the best accuracy in future launch estimates.

Thank you!